

3. Übung: Histogrammanpassung

Michael Grunwald

Aufgabe 1

- Aufgabe 1.1 Implementieren Sie eine MEX Funktion (C/C++), welche aus einem 8-Bit-Graustufenbild das zugehörige kumulative Histogramm berechnet. Als Hilfe kann der Quellcode in 'mxCumHisto.cpp' benutzt werden.
 - Prototyp: `[h] = mxCumHisto(image, binSize);`
 - `image` ist ein 8-Bit-Grauwert Bild (`image = imread('image.jpg');`);
 - `binSize` ist die Bin-Größe (bin size). Ist die Bin-Größe gleich eins (zwei, ..., 256) werden 256 (128, ..., 1) Grauwertklassen verwendet. `imread('image.jpg');`);
- Aufgabe 1.2 Laden Sie das Buch Digitale Bildverarbeitung herunter und lesen Sie Kapitel 5 (*Punktoperationen – 5.2 Punktoperationen und Histogramme*) und Kapitel 6 (Filter – 6.1 - 6.3). Bitte beachten Sie, dass der Link nur intern im HTWG Netz funktioniert.

Aufgabe 2

Beantworten Sie folgende Fragen.

- Aufgabe 2.1 Was ist ein Filter?
- Aufgabe 2.2 Was ist der Unterschied zwischen Punktoperationen und Filteroperationen?

Aufgabe 3

- Implementieren Sie eine MEX Funktion (C/C++), welche ein Bild an ein anderes Bild angleicht. Der Angleich soll über die Methode der Histogrammanpassung gelöst werden.
 - * `[LUT] = mxHistoAdapt(refHisto, adaptHisto)`
 - * `refHisto`
Histogramm des Referenzbildes
 - * `adaptHisto`
Histogramm des anzupassenden Bildes
 - * LUT
Die *Lookup* Tabelle, welche auf das anzupassende Bild angewendet, die Histogrammanpassung durchführt.

3. Übung: Histogrammanpassung

Michael Grunwald

Bemerkung

- Die beiden Eingangshistogramme in Aufgabe 3 müssen normiert sein.
- Bitte erstellen Sie für die verwendeten cpp-Dateien die zugehörigen kommentierten Matlab m-Dateien.
- Die Aufgaben werden elektronisch (m.grunwald@htwg-konstanz.de) und per Ausdruck abgegeben. (Ausdruck: mxCumHisto.cpp, mxHistoAdapt.cpp und Beantwortung der Aufgabe 2, Elektronisch: Alle cpp-Dateien und Aufgabe 3 als pdf-Datei)